IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

(Attorney Docket No. 13286:18)

TITLE

SYSTEM AND METHOD FOR PREVENTING SOFTWARE PIRACY

INVENTORS

Ravi Verma
14500 Dallas Parkway #2189
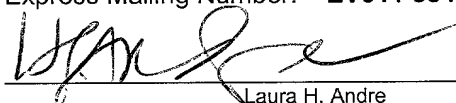**Dallas, Texas 75254**

ASSIGNEE

Networks Associates Technology, Inc.
3965 Freedom Circle
Santa Clara, CA 95054

SYSTEM AND METHOD FOR PREVENTING SOFTWARE PIRACY

TECHNICAL FIELD OF THE INVENTION

[0001]   This invention relates in general to a system and method for preventing unauthorized use of software. In particular, the invention relates to a method of validating licensed activities through an interconnected network.

BACKGROUND OF THE INVENTION

[0002]   Software developers build revenue streams by licensing software to end users. However, significant losses in revenue may occur when software is pirated. As such, unlicensed use represents a large revenue loss to software companies. However, there is no unified method of preventing piracy.

[0003]   In one exemplary licensing scenario, software products are licensed to companies. The license may be for a multitude of instances of the software application which may expire on a given expiration date. For example, a company may license ten copies of an application that expire at the end of the year. In this example, the revenue model for the software licensor is circumvented when more than the licensed number of copies are used or the copies are used beyond the expiration date.

[0004]   In one typical embodiment, a server delivers the applications to a computer connected to the network. As such, the server keeps track of the number of applications delivered or the number of instances of the application in service. However, this method may be circumvented by detaching the computer from the network once the application has been served. Alternately, the server may lose track of the number of applications delivered, especially in the event of network failures or other network events.

[0005]   A further typical solution is to require a license server to be named upon installation. This method may suffer similar deficiencies as the method above. Further, customers may be reluctant to undergo the expense associated with installing and maintaining a license server.

[0006]   In another situation, applications are often sold or licensed to individuals for use on a single machine. These machines may be periodically connected to a network. A problem exists when one copy is installed on multiple machines.

[0007]   Other typical solutions require a software key in order to enable the application to run. However, if the software is copied onto a second computer, the same key may be used; thus circumventing the licensing process. Such piracy accounts for a significant revenue loss to the software developer.

5   **[0008]**   Another typical solution is to require electronic registration. With this solution, a software company may determine if multiple copies of the same licensed application have been installed on various machines. The electronic registration method may also attempt to determine if other pirated software resides on the machine requesting the registration. However, this method is often circumvented by users not registering the software.

10   **[0009]**   On the other hand, companies have seen an increased liability associated with having pirated software on company computers. Periodic software audits that find pirated software may lead to costly fines. As a defensive measure, companies may be tempted to purchase more than the required number of licenses to limit their liability. In some instances, they may purchase licenses for all of the computers owned in the facility. However, the software may or may not be
15   used at all the computers simultaneously.

[0010]   As such, deficiencies exist in delivering software licenses. Software keys may be copied. Software delivered through servers may be pirated through errors in the network or the server. In addition, many other typical licensing mechanisms may be circumvented through network failure or the sharing of licensing and licensed keys.

20   **[0011]**   As such, many typical licensing mechanisms suffer from deficiencies in providing and preventing piracy of software and providing an economic means for licensing software by companies. Many other problems and disadvantages of the prior art will become apparent to one skilled in the art after comparing such prior art with the present invention described herein.

SUMMARY OF THE INVENTION

25   **[0012]**   Aspects of the invention are found in a license enforcement server. The license enforcement server may act to communicate with computers and other devices connected to a network. The communication may include requests for permission to run an application or messages indicating permission to run applications.

2

[0013]   The license enforcement server may be located on a network machine. Further, the license enforcement server may be associated with the operating system running on the network machine. In addition, the network machine may have a licensed usage database, a namespace tree and/or communications instructions associated with delivering licenses.

5   **[0014]**   Aspects of the invention may also be found in a computational device connected to the network. The computational device may have software instructions for communicating with the license enforcement server. Further, the computational device may have application software. The application software may have instruction sets for communication with the license enforcement server. In addition, the software may have instructions enabling or disabling the 10   software in response to messages from the license enforcement server.

[0015]   Further aspects of the invention may be found in an interconnected network to which the network machine and the computational devices are attached. The network machine, having the licensed enforcement server associated with the operating system running on the network machine, may be a network-critical machine. In addition, computational devices may be 15   connected to the interconnected network. The computational devices may have software instruction sets for communicating with the network-critical machine and the license enforcement server residing thereon.

[0016]   Further aspects of the invention may also be found in a secondary license enforcement server. The secondary license enforcement server may act as a back up to the first 20   license enforcement server. The secondary license enforcement server may reside on a second network-critical machine and/or a machine acting to back up the first network-critical machine. The second license enforcement server may be in communication with the first license enforcement server. Further, the second license enforcement server may maintain a back up license database, usage database and/or namespace tree.

25   **[0017]**   The network-critical machines may be machines such as domain name servers and address servers, among others. Further, these servers may use operating systems such as Microsoft Windows NT®; Microsoft Windows XP®; Microsoft Windows 2000®; Microsoft Windows 98®; MacOS 9.x®; MaxOS X®; Linux®; FreeBSD®, UNIX®, and others. The license enforcement server may run as a system critical thread associated with the operating 30   system or, the license enforcement server may be part of the operating system, among others.

[0018] Further aspects of the invention may be found in a machine-readable storage medium. The machine-readable storage medium may contain instruction sets for communicating between the license enforcement server and the application software. Further, the machine-readable storage medium may contain the application software or instruction sets operable to communicate with the license enforcement server. Alternately, the machine-readable storage medium may contain instruction sets for installing an application software. In another example, the machine-readable storage medium may contain the instruction sets of an operating system incorporating the license enforcement server.

[0019] Aspects of the invention may also be found in a method for obtaining permission to perform a licensed activity. For example, a software application may be initiated on a computational device. The software application may query the license enforcement server to obtain permission to run the application. In another example, an installation program associated with the application may query the license enforcement server to obtain permission to install the application on the computational device. The method may include steps of poling the license enforcement server, selectively determining the permissibility of the licensed activity, and performing the licensed activity.

[0020] As such, a system for preventing software piracy is described. Other aspects, advantages, and novel features of the present invention will become apparent from the Detailed Description of the Invention when considered in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] For a more complete understanding of the present invention and advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings in which like reference numbers indicate like features and wherein:

[0022] FIG. 1 is a schematic block diagram of the system according to the invention.

[0023] FIG. 2 is a schematic block diagram of an exemplary embodiment of the system as seen in FIG. 1.

[0024] FIG. 3 is a schematic block diagram of an exemplary embodiment of the system as seen in FIG. 1.

[0025] FIG. 4 is a schematic block diagram of an exemplary embodiment of a network machine for use in the system as seen in FIG. 1.

[0026] FIG. 5 is a block diagram of an exemplary embodiment of a computational device for use with the system as seen in FIG. 1.

[0027] FIG. 6 is a schematic block diagram of an exemplary embodiment of the system as seen in FIG. 1.

[0028] FIG. 7 is schematic block diagram of an exemplary embodiment of data structure for use in the system as seen in FIG. 3.

[0029] FIG. 8 is a block flow diagram of an exemplary method for use by the system as seen in FIG. 1.

[0030] FIG. 9 is a block flow diagram of an exemplary method for use by the system as seen in FIG. 3.

[0031] Corresponding reference numerals indicate corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0032] FIG. 1 is a schematic block diagram of the system according to the invention. The system 10 has a license enforcement server 12 connected with a computational device 14 through a communications means 16. The device 14 may have a licensed software component. This software may query the license enforcement server 12 through the communication means 16 to determine the status or permissibility of the license or licensed activity.

[0033] A licensed enforcement server 12 may take various forms. For example, the license enforcement server may take a hardware form such as a network device or server. Alternately, the license enforcement server may be a software component residing on a server or network device. However, the license enforcement server 12 may take these forms or various combinations of these forms and may be a combination of hardware and software.

[0034] A server and/or network device may take various forms. These forms may include an email server, a primary domain control server, a backup domain control server, a backbone server, a server providing DNS service or a server providing DHCP service, among others. Alternately, the license enforcement server 12 may be associated with switches and hubs. Moreover, the server and/or network device may or may not be a network-critical server and/or network device. A network-critical server and/or network device may be a server and/or

network device, the failure of which would disable and/or diminish the functionality of the network. The functionality may, for example, be an essential functionality, the general operability of the network, or an addressing service, among others.

[0035]   A licensed activity may take various forms. These forms may include the installation of software, running an application, accessing components of an application, accessing functionality of a software, and performing functions, among others.

[0036]   In one potential embodiment, the license enforcement server 12 may be an integral part of the operating system associated with a server or network device. The operating system may take various forms. These forms may include Windows 98®, Windows NT®, Microsoft Windows 2000®, Microsoft Windows XP®, Linux®, BSD®, UNIX®, MacOS9.X®, MacOSX®, and others.

[0037]   In one exemplary embodiment, the license enforcement server 12 may reside on a server acting as the primary domain control server, a domain name server, or an address server, among others. A computational device connected to a network 16 may interact with the license enforcement server to determine whether an activity is licensed. The licensed activity may, for example, be the installation of a software application or the continued running of a software application, among others.

[0038]   Alternately, a computational device 14 may periodically be connected to a network 16. When connected, the computational device 14 may confirm permission to perform a licensed activity with a license enforcement server 12. The license enforcement server 12 may take the form, for example, of a domain server or license server connected to a global network. Further, the licensed activity may be the installation of software for which, a license is confirmed, obtained, purchased, and/or acquired, among others, through a network connection.

[0039]   As such, various licensing models may be enabled. For example, a limited date model may be used in which a license is provided with a termination date. Alternately, this model could be used to license software for a period of time from the date of installation. For example, a license may be valid to a specific date or, alternately, three months from installation.

[0040]   Another license model might be the limited launch model. In this model the license may be expressed in a number of application launches. For example, each time an instance of the application is launched, an available launch count is decremented. Once the available launch count reaches zero, the license would expire.

[0041]   In another exemplary model, the license may be expressed in an amount of time. For example, the license may permit 200 hours of operation. This license may be accomplished by periodic querying of the license server. Alternately, the application may provide a beginning and ending time or some other time difference.

5   **[0042]**   Another licensing approach would be a limited instance model. In this model, a finite number of instances of the application would be allowed at any one time. However, various combinations of these models may be envisaged. For example, an application may be licensed for a specific date or period with a finite number of installations. In another exemplary embodiment the license may be expressed in terms of a finite amount of application running time

10   with a limited number of launches.

[0043]   In this manner, software developers and software companies may prevent piracy using the license server system above. In addition, companies may benefit by more optimally allocating licensing costs over time. Further, these companies may benefit by limited exposure to fines and penalties associated with having unlicensed software.

15   **[0044]**   FIG. 2 is a schematic block diagram of an exemplary embodiment of the system as seen in FIG. 1. The system 30 may have an interconnected network 32 to which at least one license enforcement server 34 may be connected. In addition, various computational devices 36, 38, 40 and 42 may be connected to the interconnected network 32. The devices 36, 38, 40 and 42 may have software components or applications that may communicate through the

20   interconnected network 32 with the license enforcement server 34 to obtain permission to perform a licensed activity. Various applications may be of different vendors or the same vendor, among others. Further, the applications may have varying functionalities, as well as varying versions of the same application.

[0045]   The computational devices may take various forms. These forms may include

25   computers, desktop computers, laptop computers, portable computers, workstations, handheld devices, smart phones and network servers, among others.

[0046]   In addition, the software may take varying forms. These forms may include applications, libraries, operating systems, wizards, and macros, among others. Further, the software may be compiled from code written in C++, Visual Basic, FORTRAN and others.

30   Alternately, the software may be interpreted from code written in Java, PERL, and Visual Basic, among others.

[0047]   The software and/or applications may obtain permission to perform a licensed activity from the license enforcement server 34. In one exemplary embodiment, the license enforcement server may reside on a server or network device. These devices may take various forms. These forms may include an email server, a primary domain control server, a backup domain control server, a backbone server, a server providing DNS service, or a server providing DHCP service, among others.

[0048]   Further, the license enforcement server may reside on a network-critical server, device or machine. The license enforcement server 34 may form an integral part of the operating system. As such, tampering or disabling the license enforcement server 34 may disable, disrupt or cause the interconnected network 32 to fail. The interconnected network 32 may take varying forms. These forms may include global networks, wide area networks, local area networks, wireless networks, and rings, among others.

[0049]   With further reference to Figure 2, a backup license enforcement server 46 may be connected to the interconnected network 32. In one exemplary embodiment, the backup license enforcement server 46 may communicate with the primary license enforcement server 34 to maintain a backup license and usage database. Further, the license enforcement server 46 may maintain a backup data associated with licenses and licensing models. In the event, that the primary license enforcement server 34 is to fail, the backup license enforcement server 46 may assume the license enforcement role with little or no loss in network and/or license enforcement service. However, the license enforcement server 46 may also act to license a different set of licensed activities and/or various combinations of licenses.

[0050]   In one exemplary embodiment, a software A residing on a device 38 may communicate with the license enforcement server 34 through the interconnected network. In doing so, the software A may acquire permission to run. In addition, the software A residing on a device 40 may communicate through the interconnected network 32 to the license enforcement server 34. Depending upon the model and the number of instances of software A running on devices connected to the interconnected network 32, software A may or may not be given permission to run.

[0051]   Software B may be installed on a device 36. Software B may be developed by a differing organization or may be, for example, a different version of a similar application to that of software A. During installation, device 36 may communicate with the license enforcement server 34 through the interconnected network 32 and obtain permission to install software B.

[0052] In another exemplary embodiment, a software C may be running on a device 42. The software C may communicate with the license enforcement server 34 through the interconnected network 32 periodically to establish that a permissible run time has not expired.

[0053] In this manner, multiple applications from varying vendors may run using varying licensing models and all obtaining licenses from a single license enforcement server and/or a combination of license enforcement servers.

[0054] In another exemplary embodiment, the license enforcement server or servers may be connected to an external network or communications means. The communications means may take various forms. These forms may include a modem, wireless connection, cable modem, Ethernet, phoneline, T1, or T3, among others.

[0055] FIG. 3 is a schematic block diagram of an exemplary embodiment of a network machine for use in the system as seen in FIG. 1. In this exemplary embodiment, the system 30, similar to that of FIG. 2, may operate such that devices may obtain permission to perform licensed activities from a license enforcement server 34 through an interconnected network 32. In one exemplary embodiment, the license enforcement server 34 and 46 may be connected to a global network 48 or another communications means. Through this communications means or global network 48, the license enforcement server 34 or 46 may obtain additional licenses for software. Alternately, a software provider may query the machine or install new licenses, among others.

[0056] Referring again to FIG. 3, in another exemplary embodiment, a licensing organization may communicate with the license enforcement server 34, or 46 through the global network 48 or other communications means. The license enforcement server 34, 46 may communicate with a license enforcement gateway 49. In this manner, the licensing organization may establish the universal licensing structure, such as, for example, a namespace tree.

[0057] With reference to FIG. 3, in another exemplary embodiment, a license enforcement gateway 49 is communicatively coupled to the global network 48 through a communications means. The communications means may take various forms. These forms may include a modem, wireless connection, cable modem, Ethernet, phoneline, T1, or T3, among others. In this exemplary embodiment a master database 47 is on or accessible through the license enforcement gateway 49.

[0058]   Periodically, a license enforcement server, such as license enforcement server 34, can connect to the license enforcement gateway 49 to synchronize and validate its own licensing database and/or namespace tree according to this particular exemplary embodiment.

[0059]   As such, a universal method for uniquely identifying licensed products may be
5   established. Further, a universal method for license delivery and validation may be established. In addition, a method for detecting, discovering, alarming, warning, and preventing, among others, unlicensed activities may be envisaged.

[0060]   The license enforcement server 34 may communicate with the license enforcement gateway 49 information relating to, among other things, for example, a particular
10   license or list of licenses, communications regarding use, queries and/or requests for new licenses and/or addendums to existing licenses, alarms and/or warnings of misuses or piracy, alarms and/or warnings of tampering or disabling, and other information based upon activity of the devices and servers operating in the system 30, among others.

[0061]   For the license or list of licenses, if a license is valid, then the license gateway 49 may
15   return to the license enforcement server 34 a "Yes", for example, to allow installation and execution of an application or software on a client or device. If a license is not valid, then the license gateway 49 may return to the license enforcement server 34 a "No", for example, to prevent installation and execution of an application or software on a client or device. Thus, when a client or device attempts to install and/or execute an unlicensed application, such client
20   or device can be stopped or notified. In this exemplary embodiment, no two clients on the Internet may use the same license if it is not permitted. However, the license enforcement server may periodically confirm a license or confirm the license in response to an event such as a date, number of launches, or number of logged hours, among others. Between confirmations, the license enforcement server may authorize licenses without accessing the license enforcement
25   gateway 49.

[0062]   By means of the global network 48 the license enforcement server 34, 46 may communicate with the license enforcement gateway 49 to update its own database based on the master database 47 within or accessible through the license enforcement gateway 49. Similarly, the license enforcement gateway 49 may be accessible to a stand-alone user.

[0063] Organizations may update the master database 47 to reflect purchases and sales activity such that databases on the license enforcement servers 34, 46 in the system 30 illustrated in FIG. 3 are kept current.

[0064] Alternately, multiple license enforcement gateways 49 may be connected to a global network. These license enforcement gateways 49 may have a listing of authorized licenses 47. In this manner, a universal method may be used to communicate with multiple vendors for obtaining licenses and/or addendums to existing licenses. However, the various vendors may maintain separate authorized licenses listings 47.

[0065] The license enforcement server may reside on a network machine. FIG. 4 is a block diagram of an exemplary embodiment of a network machine for use in the system as seen in FIG. 1. The network machine 50 may have a processor 52, a programmable circuitry 54, a network interface 56, and one or more storage mediums 58. The processor 52 and/or the programmable circuitry 54 may access the storage medium or mediums 58 to acquire instructions associated with the operating system 66. Further, the network machine 50 may combine these various components and various configurations. However, many other configurations can be envisaged.

[0066] Associated with the operating system may be the license enforcement server 68 or components thereof. Alternately, the license enforcement server 72 may be separate from the operating system.

[0067] The processor may take many various forms. These forms may include various microprocessors, chips, and processing apparatuses, among others. Further, the system may have more than one processor.

[0068] The programmable circuitry may take various forms. These forms may be adaptable or programmed through various means. These means may include keyboards, graphical user interfacing devices, handheld circuitries, across networks, and others.

[0069] The network interface 56 may take various forms. These forms may connect the network interface 56 to a local area network, wide area network, or global network, among others. Further, the network interface 56 may function to communicate with computational devices seeking permission to perform license activities, other network machines, network machines with back up license enforcement servers, software licensors, and software organizations, among others. Further, the network interface may use various protocols to

11

communicate. These protocols may include TCP/IP, Ethernet protocols, licensing protocols and others.

[0070]   Storage medium or mediums 58 may take varying forms. These forms may include RAM, ROM, flash memory, hard drives, floppy drives, removable drives, CDs, DVDs, memory stick, and memory cards, among others.

[0071]   A storage medium or mediums 58 may hold communications instructions 60, a licensing database 62, usage database 64, an operating system 68, and a namespace tree 70, among others. Further, a license enforcement server 68 may be associated or integrated into the operating system 66. Alternately, a license enforcement server 72 may be stored separate from the operating system 66. However, various parts, components, subroutine, and libraries, among others may or may not, in various combinations, be associated with the operating system.

[0072]   The communications instructions 60 may enable the network machine to communicate with various computational devices seeking licenses to perform licensed activities. A licensed database 62 may be a database of licenses purchased for various products and current usage accounts associated with those products. Alternately, a usage database may store the information associated with current usage or information associated with various methods for licensing software. These databases may take varying forms. These forms may include text files, spreadsheets, Microsoft Jet databases, My SQL relational databases, and object databases, among others.

[0073]   Further, the databases may be organized using a namespace tree 70. The namespace tree 70 may uniquely identify various organizations, products, and versions associated with software applications. In addition, the namespace tree may be maintained by a single organization and may be communicated to the network machine through a global network or other communications means.

[0074]   In one exemplary embodiment, the network machine may be a machine critical to network performance. For example, the network-critical machine may take the form of an email server, a primary domain control server, a backup domain control server, a backbone server, a server providing DNS service, or a server providing DHCP service, among others.

[0075]   In the case where the license enforcement server or components thereof 68 are integrally tied to the operating system 66. An attempt to disable the license enforcement server 68 may cripple or disable the network. In this manner, piracy may be prevented or discouraged.

[0076] FIG. 5 is a schematic block diagram of an exemplary embodiment and a computational device as seen in FIG. 1. The computational device 90 may have a processor 92, a programmable circuitry 94, a network interface 96, and one or more storage mediums 98. The storage medium or mediums 98 may have software instruction sets associated with an operating system 100, a license API 102, a software A 104, a software B 108, a software C 112 and/or an installation client 114. The computational device may have some, all or none of these components. Further, the computational device 90 may combine these various components and various configurations. However, many other configurations can be envisaged.

[0077] The processor 92 may take various forms. These forms may include various microprocessors, chips, chipsets, and other processing apparatuses.

[0078] The programmable circuitry 94 may take various forms. These forms may be adaptable or programmed through various means. These means may include keyboards, graphical user interfacing devices, handheld circuitries, across networks, and others.

[0079] The network interface 96 may take various forms. These forms may connect the network interface 56 to a local area network, wide area network, or global network, among others. Further, the network interface 56 may function to communicate with network machines with primary or back-up license enforcement servers, and other network machines, among others. Further, the network interface may use various protocols to communicate. These protocols may include TCP/IP, Ethernet protocols, licensing protocols and others.

[0080] The storage medium or mediums 98 may take various forms. These forms may include RAM, ROM, flash memory, hard drives, floppy drives, removable drives, CDs, DVDs, memory stick, and memory bands, among others.

[0081] The storage medium or mediums 98 may have a license API 102 comprising instructions for communication with the license enforcement server. Further, the license API 102 may form an interface with the license enforcement server and various software. In one example, the license API 102 may take various forms such as dynamic linked libraries, components, and/or threads of the operating system, and executables, among others.

[0082] In another exemplary embodiment, the license API 106 may be incorporated into a software A 104. In this manner, a license API 106 may enable software A to communicate with the license enforcement server through the network interface 96. Alternately, an application client 110 may be incorporated in a software B 108. The application client 110 may interface

with the license API 102 to communicate through the network interface 96 with the license enforcement server.

[0083]   In another exemplary embodiment, an installation client 114 may communicate with the license enforcement server using the license API 102, or on its own. In either case, the

5   installation client may communicate with the license enforcement server through the network interface 96. In this case, the installation client may seek permission to install software C 112.

[0084]   In this manner, various licensing models may be used. Further, these licensing models may be employed on various software programs residing on the same computational device 90. However, some, all or none of these licensing models may be in use on the

10   computational device 90. In addition, various combinations of licensing models may be used or incorporated into the various software products associated with the computational device 90.

[0085]   In another exemplary embodiment, sets of client machines are connected through an interconnected network to a network-critical server. FIG. 6 is a schematic block diagram of another exemplary embodiment of the system as seen in FIG. 1. In FIG. 6, sets of clients

15   communicate with the network-critical server using a communications protocol ("CP"). The protocol may be a standard protocol or one specific to a licensing enforcement server method. The network-critical server denoted the primary server ("PS") has the license enforcement server as an integrated part of the operating system. When prompted for permission to perform a licensed activity, the license enforcement server may communicate with the licensed database,

20   the usage database or another database to ascertain the availability of a license.

[0086]   A backup server ("BS") may communicate with the primary server through a similar communications protocol or a protocol specific to the license enforcement server method. Through this interaction, the backup server may create a backup license database, usage database, or other data associated with the license enforcement server. If the primary server

25   were to fail, the backup server may take on the role of the primary server.

[0087]   In one exemplary embodiment, the primary server may be a primary domain control server. The license enforcement server may be an integrated part of the operating system of the primary domain control server. A backup domain control server may similarly have a license enforcement server software incorporated into the operating system. The backup domain control

30   server may communicate with the primary domain control server to acquire data, databases, licensed databases, usage databases, and other data associated with the license enforcement

server. If the primary domain control server were to fail, the backup domain control server would activate and take over the role of the license enforcement server.

[0088]   With this system, one or more client computational devices may interact with a network-critical server to obtain permission to perform a licensed activity. Further, piracy would

5   be limited by the fact that disabling of the primary control server would either disable the network preventing permission to be delivered for performing the licensed activity or activate the backup server, maintaining the network and the license enforcement server mechanisms. In addition, various client computational devices may interact with the license enforcement servers. Further, various software produced by various vendors may interact simultaneously with the

10   license enforcement servers.

[0089]   One aspect of the invention may be found in the namespace tree. In order to organize various software products produced by various vendors and serve licenses using various licensing models, the namespace tree provides a means of organization.

[0090]   FIG. 7 is a schematic block flow diagram of an exemplary embodiment of a

15   namespace tree for use in the system as seen in FIG. 1. . In order to organize various software products produced by various vendors and serve licenses using various licensing models, the namespace tree provides a means of organization.

[0091]   The namespace tree may be organized using data associated with vendors or organizations, categories of software, products, and versions of those products, and licensing

20   modeled data associated with those products, among others. The namespace tree may be organized in the hierarchical fashion as denoted in FIG. 7. However, various other organizations may be envisaged.

[0092]   With the namespace tree, an organization such as an international standards organization or a vendor of the license enforcement server may establish a universal license

25   enforcement database structure. With this universal license enforcement database structure, one license enforcement server may serve licenses associated with software from various vendors, various products, and various versions.

[0093]   FIG. 8 is a block flow diagram of an exemplary method for use by the system as seen in FIG. 1. The method 130 may begin with an initiated action 132. This action may be a user

30   initiating the start of a software application or, for example, the user initiating an installation of a software program. However, this step may or may not be included in the method.

[0094]   In a next step, as denoted in block 134, a client computational device may pole the license enforcement server requesting permission to perform a licensed activity. The licensed activity may, for example, be the installation of a software, the running of a software or the continued running of a software, among others.

5   **[0095]**   In a further step as denoted in a block 136, the license enforcement server may determine permissibility. The license enforcement server may use various information to determine whether a licensed activity is permissible. For example, the license enforcement server may use the IP address of the client machine, the type of software, the version of software, the number of similar applications running on the network simultaneously, the date, the amount

10   of time that application or similar applications have been running and the number of application launches, among others. Once the license enforcement server has determined permissibility, it may send a message to the client computational device denoting the permissibility.

[0101]   As seen in block 138, if the activity is permissible, the client machine may allow access as seen in block 140. However, if permission is denied, the client machine may disallow

15   access as seen in block 142.

[0102]   The license enforcement server may deny permission in the event that a license has expired or the application count is exceeded, among other reasons. FIG. 9 is a block flow diagram of an exemplary method for use in the system as seen in FIG. 3. The method may be used to obtain a license, additional license, addendum to a license, or confirm a license, among

20   others. Further, other method steps may be envisaged to perform these and other activities.

[0103]   In a block 152, the license enforcement server may receive a request for permission to perform a licensed activity. The request may have been generated by, for example, a client machine. However, the request may be generated through other means. Further, this step may be replaced by a query from a license enforcement gateway. Alternately, the step as seen in

25   block 152 may or may not be included in the method.

[0104]   In another exemplary embodiment, the method may seek to determine the status of the license periodically. As such, the step as seen in the block 152, may not be included in the method.

[0105]   Returning to FIG. 9, in response to receiving the request or other initiating steps, the

30   licenses enforcement sever may determine the status of the license, as seen in a block 154. If the license is available, the license enforcement server may selectively authorize the licensed

activity, as seen in a block 160. This authorization may, for example, include responding to the request for permission.

[0106]   If, however, the license is not available or expired, the license enforcement server may request a new license, an addendum to an existing license, or an extension, among others, as

5   seen in a block 156. For example, a time limit may have been exceeded by previous uses of a licensed application. The license enforcement server may be configured to acquire an extension of the time. Alternately, an expiration date may have passed and the license enforcement server may acquire a new expiration date. Other embodiments may include adding more to the permitted application count, increasing the number of permitted launches, and increasing the

10   number of permitted installations, among others.

[0107]   An additional step may or may not be incorporated requiring user interaction to permit the request. Further various algorithms may be used in determining whether to make a request for a change in licenses. Moreover, various means may be established for payment for the contract. In addition, the license enforcement server may have means for establishing a

15   temporary license.

[0108]   To request additional licenses or change existing licenses, the license enforcement server may communicate with a license enforcement gateway or other system through a global network. However, other means may be envisaged for requesting a license. The license enforcement server may, through the communication or other means, determine the availability

20   of a license, as seen in a block 158. This determination may be made by receiving updated, new, or additional information and/or data regarding the license. The license enforcement server may then adjust the availability of the license.

[0109]   If permission is established, the license enforcement server may authorize the licensed activity as seen in the block 160. However, if an addendum, change, new, or

25   supplemental license is not established, the license enforcement server may deny permission to perform the licensed activity as seen in a block 162.

[0110]   In this manner, the license enforcement server may automatically update licenses. Further through the communication with the license enforcement gateway, the license enforcement server may inform the licensor of attempts to perform licensed activities without a

30   license. Moreover, the method may enable a company to more optimally license software based

17

on smaller increments of use. As such, this approach may be more economical for both the licensee and licensor. The method and system may prevent both accidental misuses and piracy.

[0111] Although the present invention has been described in detail, it should be understood that various changes, substitutions and alterations can be made hereto without departing from the

5   spirit and scope of the invention as defined by the appended claims.